

DOCUMENT RESUME

ED 116 607

IR 002 471

AUTHOR Nuttall, Herbert E., Jr.; Himmelblau, David M.
 TITLE Interactive Reactor Simulation.
 INSTITUTION Texas Univ., Austin. Project C-BE.
 SPONS AGENCY National Science Foundation, Washington, D.C.
 REPORT NO EP-10-05-07-73
 PUB DATE 7 May 73
 NOTE 40p.; For related documents see IR 002 463 and 464

EDRS PRICE MF-\$0.76 HC-\$1.95 Plus Postage
 DESCRIPTORS Chemistry; *Computer Assisted Instruction; Computer Graphics; Computers; Course Evaluation; *Engineering Education; Engineering Graphics; Engineering Technology; Higher Education; Input Output Devices; Models; *Programming; *Simulation; Student Attitudes; Technical Education
 IDENTIFIERS Chemical Engineering; DYFLO; Dynamic Process Simulation; *Interactive Programs; Project C BE

ABSTRACT

In the field of chemical engineering, interactive process models can simulate the dynamic behavior and analysis of chemical processes. DYFLO was the process simulation program selected as a foundation for development of interactive programs for computer-assisted instruction (CAI) in chemical engineering. Interactive Computing and time sharing graphics allowed students to assess and explore the dynamic behavior of simulated processes. Classroom CAI applications of these interactive programs showed favorable student attitudes toward computer lessons as a supplement to regular class presentations. (CH)

 * Documents acquired by ERIC include many informal unpublished *
 * materials not available from other sources. ERIC makes every effort *
 * to obtain the best copy available. Nevertheless, items of marginal *
 * reproducibility are often encountered and this affects the quality *
 * of the microfiche and hardcopy reproduction. ERIC makes available *
 * via the ERIC Document Reproduction Service (EDRS). EDRS is not *
 * responsible for the quality of the original document. Reproductions *
 * supplied by EDRS are the best that can be made from the original. *

ED116607

INTERACTIVE REACTOR SIMULATION

EP-10/05/07/73

Herbert E. Nuttall, Jr.
David M. Himmelb'au
Department of Chemical Engineering
The University of Texas, Austin, Texas 78712

This study was supported in part by
Project C-BE under Grant CY-9340
"The Use of Computer-Based Teaching
Techniques In Undergraduate Science
and Engineering Education," from
the National Science Foundation to
The University of Texas at Austin,
Drs. John J. Allan and J. J. Lagowski,
Co-Directors.

IR 002 471

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRO-
DUCED EXACTLY AS RECEIVED FROM
THE PERSON OR ORGANIZATION ORIGIN-
ATING IT. POINTS OF VIEW OR OPINIONS
STATED DO NOT NECESSARILY REPRESENT
OFFICIAL NATIONAL INSTITUTE OF
EDUCATION POSITION OR POLICY.

INTERACTIVE REACTOR SIMULATION

by

Herbert E. Nuttall, Jr.

David M. Himmelblau

Department of Chemical Engineering

The University of Texas, Austin, Texas 78712

SUMMARY

Currently at the University of Texas a number of departments are using time-share computing as an aid in teaching a variety of technical concepts. In the field of chemical engineering several interactive process models were developed and are being used to instruct students in the dynamic behavior and analysis of chemical processes. Interactive computing and time-share graphics were combined to provide students with a unique opportunity to assess and explore the dynamic behavior of simulated processes. The development and classroom application of these interactive programs are discussed along with the programming methods and the necessary time-share equipment.

INTRODUCTION

Process modeling and digital computer simulation have been areas of active research in chemical engineering for the last ten years. Recent advances in both computer hardware and software have added a new dimension to the development and application of simulation models. Combined with appropriate graphical displays, time-sharing facilities make it possible

to provide the illusion of operating a physical process. At The University of Texas at Austin, time-share computing has evolved into a powerful tool commonly used in teaching a wide variety of technical concepts.

Two of the dynamic process simulation programs currently under development by the Chemical Engineering Department are described. This work is part of an interdisciplinary computer-based education program sponsored by Project C-BE at The University of Texas and funded by the National Science Foundation under Grant GY-9340.

Interactive computer programs were developed which simulate the dynamic behavior of standard chemical processing equipment and provide the user with both the experience of operating simulated equipment and the opportunity to change model parameters or equipment design specifications during the simulation. Both the peripheral time-share equipment and the software used in this phase of the project were somewhat restricted to aid the transferability of the programs to other educational institutions.

Interactive programs for the continuous stirred-tank reactor and the tubular reactor have been used in conjunction with homework assignments for senior courses in process simulation and process control. The CSTR and tubular reactor programs were developed so that the user can change the process parameters at will and immediately view, both graphically and numerically, the process response.

Early efforts in chemical process simulation were focused on the steady state simulation of large and often complex processes. These programs,

such as Pacer (1), CSMP (2), FLOWTRAN (3), CHESS (4), and many others have evolved and been useful to industry and educational institutions. More recently dynamic process simulation programs have been developed such as DYNSSYS (5), DYFLO (6), PRODYC (7), and REMUS (8). To meet our objectives of developing interactive dynamic simulation programs we used DYFLO as a foundation and added user interaction and computer graphics. DYFLO is readily available and extensively documented in R. Franks' (6) book.

SIMULATION MODELS

Two unit operations, the continuous stirred-tank reactor and the tubular reactor, were used as process models in the simulation programs because they so frequently appear in text books (9,10) to illustrate a variety of chemical engineering subjects including process dynamics, control theory, and chemical kinetics. Students and practicing engineers are and in the future will continually be exposed to these two classical unit operations. In addition, these two process models are useful in teaching more advanced topics such as stochastic modeling and optimal control policies.

The simulated CSTR model contains a mixed tank reactor, cooling or heating coils, and a proportional-integral controller circuit. Figure 1 illustrates the complete configuration, and the equations representing the dynamic behavior of this system are:

(1) Energy balance for the reactor

$$\frac{dT}{dt} = \frac{q}{V} (T_i - T) - \frac{\Delta H_R R_A}{C_p \rho} - \frac{Q}{C_p \rho} \quad (1)$$

(2) Energy balance for the internal coils (based on arithmetic average driving temperature)

$$Q = U A \left[\frac{2q_c (\rho C_p)_c (T - T_{ci})}{2q_c (\rho C_p)_c + U A} \right] \quad (2)$$

(3) Mass balance for the reactant

$$\frac{dC_A}{dt} = \frac{q_i}{V} (C_{Ai} - C_A) - R_A \quad (3)$$

(4) Controller circuit equations

$$E_{PC} = \text{normalized error} = (T - SP) / \text{SPAN} \quad (4)$$

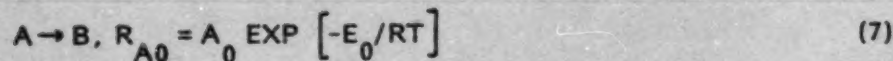
$$CO = \frac{100}{PB} (EPC + CR \int EPC dt) \Delta XN \quad (5)$$

$$q_{ci} = 60 (CO/100) + 0.1 \quad (6)$$

(5) Reaction kinetics, zero, first or second order

The user is given a choice among the three elementary reaction schemes.

These reactions were chosen because they simplify the kinetic aspects of the model.



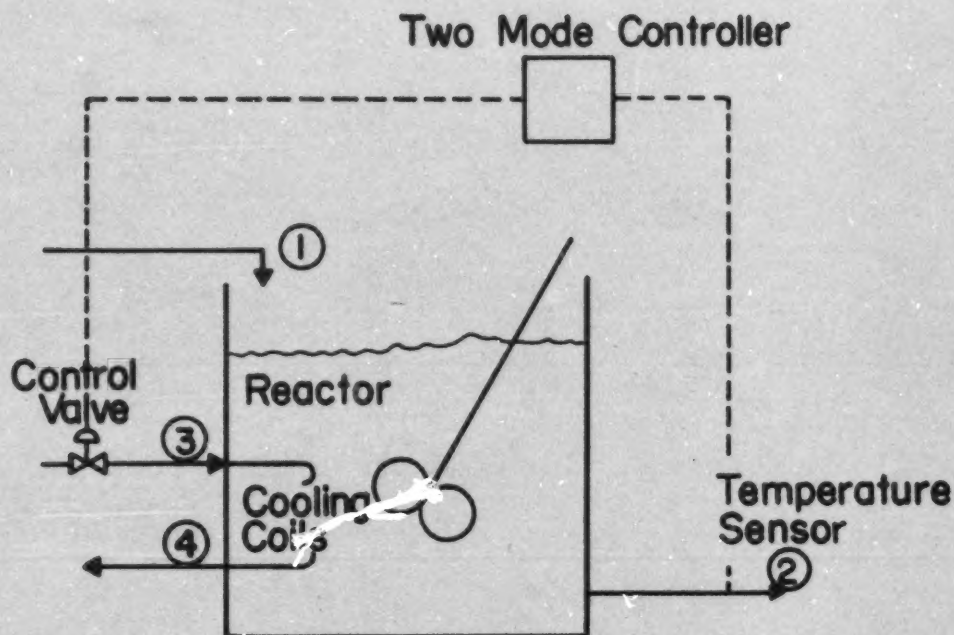


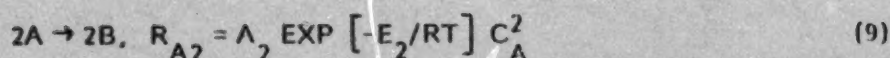
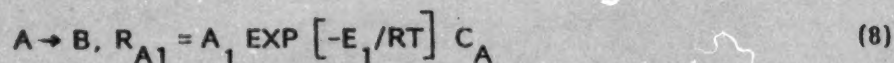
Figure 1
SCHEMATIC DIAGRAM OF THE CSTR SYSTEM.

Design Data

Reactor : 1) Constant Volume, 100 ft^3 .
2) Well Mixed.

Cooling Coils : 1) Heat Transfer Area, 500 ft^2 .
2) Heat Transfer Coefficient is independent of coolant flow rate and temperature.

Controller : 1) Measures temperature and controls the coolant flow rate.
2) Temperature range for the controller is 60°F to 210°F .
3) Linear valve on the cooling system, range 0.1 to $60.1 \text{ ft}^3/\text{min}$.



The addition of complex and more realistic reaction schemes in no way significantly limits either the model or the computer algorithm, thus improving the industrial application of the concepts and the programs developed in this work.

The tubular reactor program is composed of several different mathematical models each describing the same tubular reactor but incorporating different physical assumptions. The first model is simply a series of continuous-stirred-tank reactors with the option that the user can specify the number of mixed reactors used by the model to approximate the tubular reactor. The well-stirred tanks in series model is known to represent a dispersion model reasonably well. Again, we used Franks' (6) simulation routines and retained the same reaction kinetics as used in the continuous-stirred-tank reactor model. This version of the tubular reactor model is nonisothermal, and the user may specify or change the reactor wall temperature at any time during the simulation.

The second model is the dynamic isothermal reactor with axial dispersion, represented by the following partial differential equation :

$$\frac{\partial C_A}{\partial t} + v_z \frac{\partial C_A}{\partial z} = D_{Az} \frac{\partial^2 C_A}{\partial z^2} - R_A \quad (10)$$

where R_A is the reaction rate defined in Equations (7, 8, 9)

$$\text{I.C.} \quad t = 0 \quad 0 \leq z \leq L \quad C_A = 0$$

$$\text{B.C.} \quad z = 0 \quad t > 0 \quad V_z C_{Ai} = V_z C_A - \frac{\partial C_A}{\partial z}$$

$$z = L \quad t > 0 \quad \frac{\partial C_A}{\partial z} = 0$$

The dispersion model is used to demonstrate the effect of different Peclet numbers on the concentration profile.

The third model that is incorporated into the computer algorithm is the isothermal plug flow reactor described by:

$$\frac{\partial C_A}{\partial t} + V_z \frac{\partial C_A}{\partial z} = -R_A \quad (11)$$

Again R_A is the reaction kinetics defined in Equations (7,8,9).

$$\text{I.C.} \quad t = 0 \quad 0 \leq z \leq L \quad C_A = 0 \quad (12)$$

$$\text{B.C.} \quad z = 0 \quad t > 0 \quad C_A = C_{Ai}$$

The plug flow model is used to illustrate the limiting case in which the axial diffusion is zero or the Peclet number, $V_z L/D_{Az}$ is infinite. These three tubular reactor models are incorporated into the same computer code.

Though both the continuous-stirred-tank model and tubular reactor are relatively elementary, they are adequate to demonstrate most of the important aspects of process dynamics and control theory.

SOFTWARE

In the initial phase of formulating the interactive program the necessary DYFLO subroutines were selected for modeling the process. To decrease the ultimate storage requirements several arrays within these subroutines were reduced. A zero-initialization subroutine was developed which zeroes all arrays and replaces two of Franks' subroutines, NRCT and START. Zeroing of arrays is not necessary for those computers which perform this operation automatically, but including this subroutine in the final program assures improved transferability to other computers.

A general flow chart for the interactive program was defined. Since the computer program is formulated around Franks' DYFLO subroutines, efforts were made to use the same main program structure suggested by Franks. In fact, only minor deviations from this structure were required. The main program was divided into a series of four sections as follows:

- (1) Initialization Section
- (2) Derivative Section
- (3) Input/Output Section (includes graphics output)
- (4) Integration Section

During execution the program cycles between Sections 2, 3, and 4. At specified times within the simulation the program outputs both numerical and graphical information, then requests input from the terminal. At this point the user may enter parameter changes or introduce changes

which control the program logic, such as a change in the interrupt time interval or a request to restart the simulation at the initial conditions.

Although some computers allow the user to interrupt and change program parameters arbitrarily, in a dynamic simulation routine parameters affecting the model can be changed only at the beginning of an integration step. For our time-share system the best method for interrupting the program was to build this type of logic into the main simulation program.

Since the user wishes to monitor the simulation of the process and introduce inputs to the program at arbitrary points in the simulation time domain rather than in real time, a user controlled time increment parameter was devised to regulate the elapsed simulation time between program interrupts, i. e., input/output. By controlling the size of the simulation time increment the user has the opportunity to change parameters and monitor the system as frequently as desired.

Input to the program is achieved through a subroutine, READ, which is called at user specified times and permits parameter changes. If the user does not wish to input new parameter values, the program assigns default values to all the parameters, and the user can initiate program operation by simply typing the two character symbol GO. In general, the READ subroutine is called from the I/O section located directly after the derivative section. When a call is made to the READ subroutine, the program requests input from the terminal and from this input decides which parameters are to be updated and notes requests for changes in the program

operation such as restarting at time zero or changing the interrupt time interval. Parameter changes are entered into the program through a format read statement. To enter a parameter change the user types the corresponding two character identification symbol, an equal sign, and the new value. Input symbols and their values are temporarily stored and additional input requested until the user enters the symbol GO. Next a series of IF statements determines whether a parameter is to be updated and directs changes in the operation of the program. Input thus may be entered in random order, and if a correction in either a symbol or its value is desired, the entry is re-typed prior to the GO command. More elaborate input schemes are possible, but this very straight-forward technique has proven satisfactory for our applications.

The simplified flow chart, Figure (2), illustrates the logic structure of the CSTR program and the tubular reactor program. The I/O section located in the center of the flow diagram provides the vehicle for the interactive features of the system. At an interrupt time, graphical information is transmitted to the plotter providing a visual display of the reactor response during the last simulation time interval. Following this information is the numerical data which describe the conditions of all streams at that specific time in the simulation. Next the READ subroutine is called, and by reviewing the past performance of the reactor, the user can introduce changes which will direct the operation for the subsequent time interval.

In addition to the input subroutine READ, preparation of considerable graphics software was necessary to provide the overall interaction and

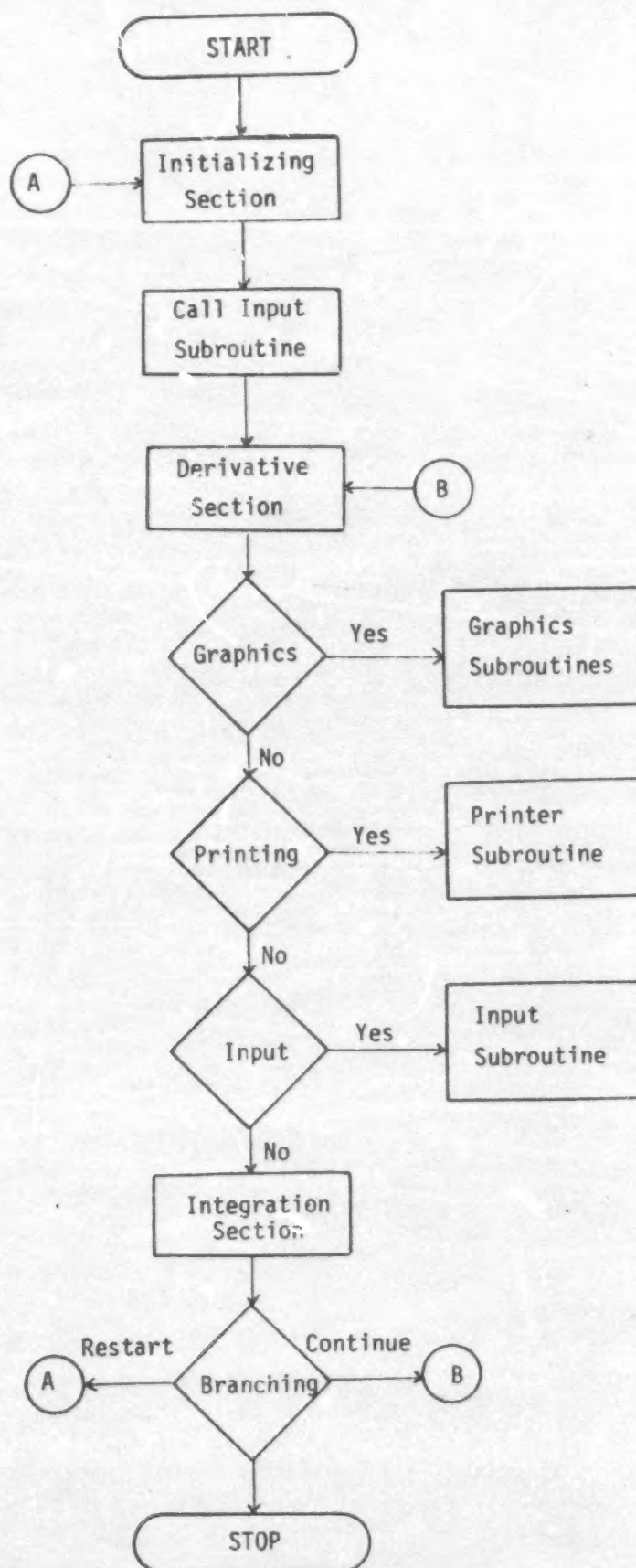


Figure 2. Flowchart for Interactive Simulation Programs

illusion of operating a process. To meet the transferability objective only standard Calcomp routines were called to perform the actual drawing of the reactor responses. A set of four plotting subroutines was developed to perform the following tasks:

- (1) Receive information from the executive routines and control the logic for all the graphics
- (2) Collect data for the plots (data is accumulated for one time interval)
- (3) Scale and draw axes
- (4) Draw reactor response curves

Though our computer center supports the batch-mode Calcomp routines, some minor additions in their time-share software were required to drive the time-share X-Y plotter with the standard Calcomp routines. This requirement in the time-share software should not degrade the transferability of the program; however, if an institution has not previously supported a time-share plotter, this software requirement should be noted.

EQUIPMENT EMPLOYED

Coupling a high speed time-share computer, an interactive terminal, and a time-share X-Y plotter, provides the user with almost immediate computational and graphical information from a dynamic simulation routine. Interactive flexibility gives the user the illusion of operating and monitoring an automated process.

Many graphs can be drawn on the same sheet of paper (12" x 17"), each graph representing a strip chart recorder operating in a plant. The terminal-plotter combination shown in Figure (3) works sequentially. Numerical information is printed or displayed at the terminal and then graphical information automatically bypasses the printer and is received by the plotter. Only one telephone line is required to connect the terminal system to the computer. An example of typical numerical output produced from a printer subroutine provided in Franks' DYFLO subroutines is shown in Figure (4). Figure (5) illustrates the corresponding graphical output from the CSTR simulation program, namely plots of the reactor temperature versus time, reactant concentration versus time, and the phase plane graph of reactor temperature versus reactant concentration. And similarly Figure (6) illustrates typical graphical output from the tubular reactor program.

The hardware used in this work consists of:

Equipment (model)	Manufacturer	Approximate Cost
Time-share computer (CDC-6400)	Control Data Corp.	
Teletypewriter (ASR-33)	Teletype Corp.	\$1000
Acoustic coupler (701A)	Omnitec Corp.	\$300
X-Y Time-share plotter (TSP-212)	Time-Share- Peripherals Corp.	\$3500

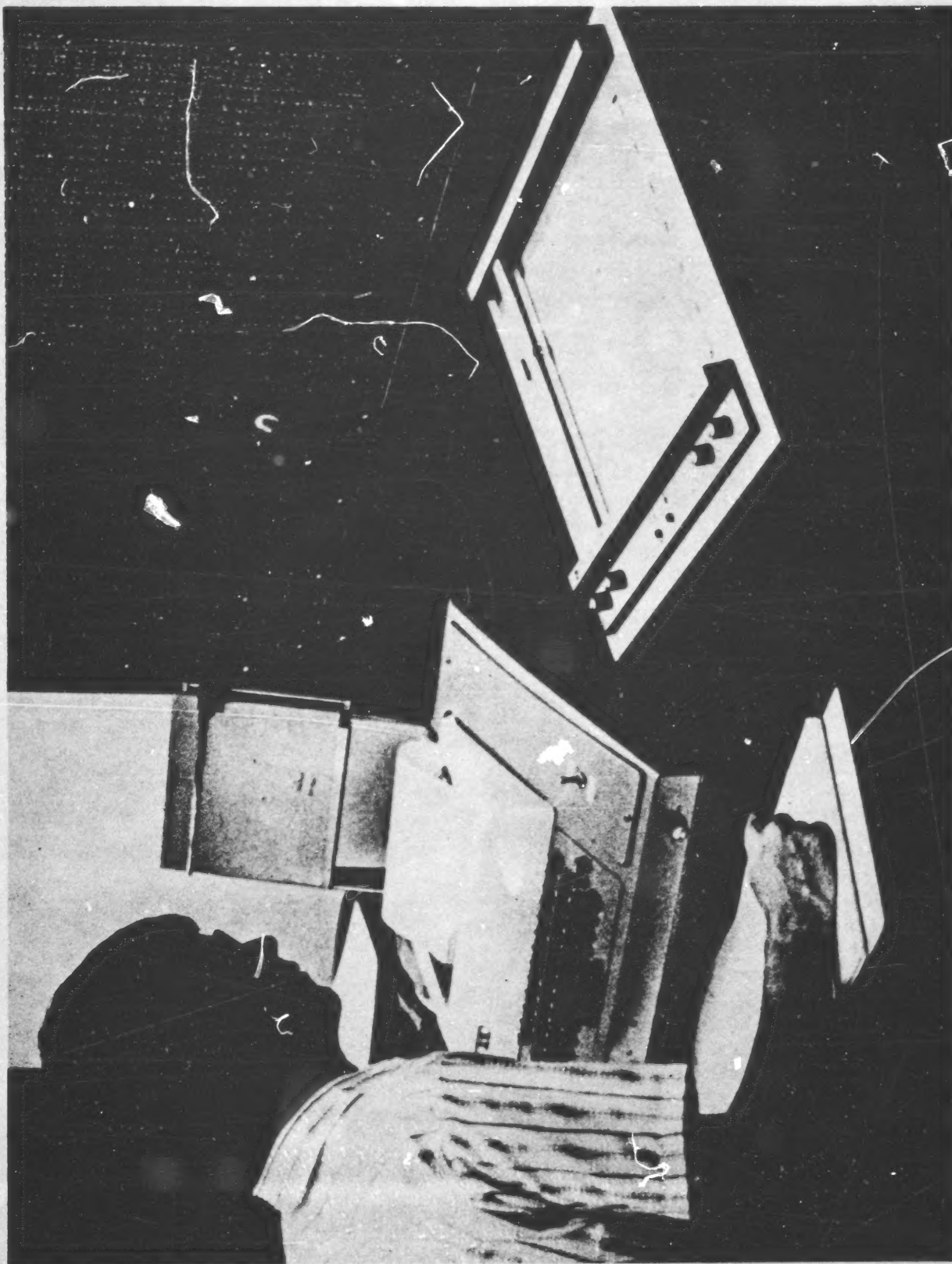


FIGURE 3. Peripheral Time-Share Equipment.

ENTER DATA

SC = 1.

GO

TIME = 0.

STRM NO	1	2	3	4
FLOW	1.800E+01	1.800E+01	8.000E+00	8.000E+00
TEMP	1.900E+02	6.000E+01	6.000E+01	6.000E+01
ENTHAL	1.190E+04	3.758E+03	3.758E+03	3.758E+03
COMP B	2.480E+00	3.480E+00	3.480E+00	3.480E+00
COMP A	1.000E+00	0.	0.	0.

TIME = 1.0000E+01

STRM NO	1	2	3	4
FLOW	1.800E+01	1.800E+01	8.000E+00	8.000E+00
TEMP	1.900E+02	1.488E+02	6.000E+01	1.407E+02
ENTHAL	1.190E+04	9.317E+03	3.758E+03	8.811E+03
COMP B	2.480E+00	2.782E+00	3.480E+00	3.480E+00
COMP A	1.000E+00	6.962E-01	0.	0.

ENTER DATA

GO

TIME = 2.0000E+01

STRM NO	1	2	3	4
FLOW	1.800E+01	1.800E+01	8.000E+00	8.000E+00
TEMP	1.900E+02	1.703E+02	6.000E+01	1.602E+02
ENTHAL	1.190E+04	1.065E+04	3.758E+03	1.003E+04
COMP B	2.480E+00	2.896E+00	3.480E+00	3.480E+00
COMP A	1.000E+00	5.785E-01	0.	0.

ENTER DATA

GO

TIME = 3.0000E+01

STRM NO	1	2	3	4
FLOW	1.800E+01	1.800E+01	8.000E+00	8.000E+00
TEMP	1.900E+02	1.758E+02	6.000E+01	1.651E+02
ENTHAL	1.190E+04	1.099E+04	3.758E+03	1.034E+04
COMP B	2.480E+00	2.989E+00	3.480E+00	3.480E+00
COMP A	1.000E+00	4.860E-01	0.	0.

ENTER DATA

GO

TIME = 4.0000E+01

STRM NO	1	2	3	4
FLOW	1.800E+01	1.800E+01	8.000E+00	8.000E+00
TEMP	1.900E+02	1.754E+02	6.000E+01	1.648E+02
ENTHAL	1.190E+04	1.097E+04	3.758E+03	1.032E+04
COMP B	2.480E+00	2.991E+00	3.480E+00	3.480E+00
COMP A	1.000E+00	4.837E-01	0.	0.

ENTER DATA

GO

Figure 4. Sample Numerical Output from CSTR Program

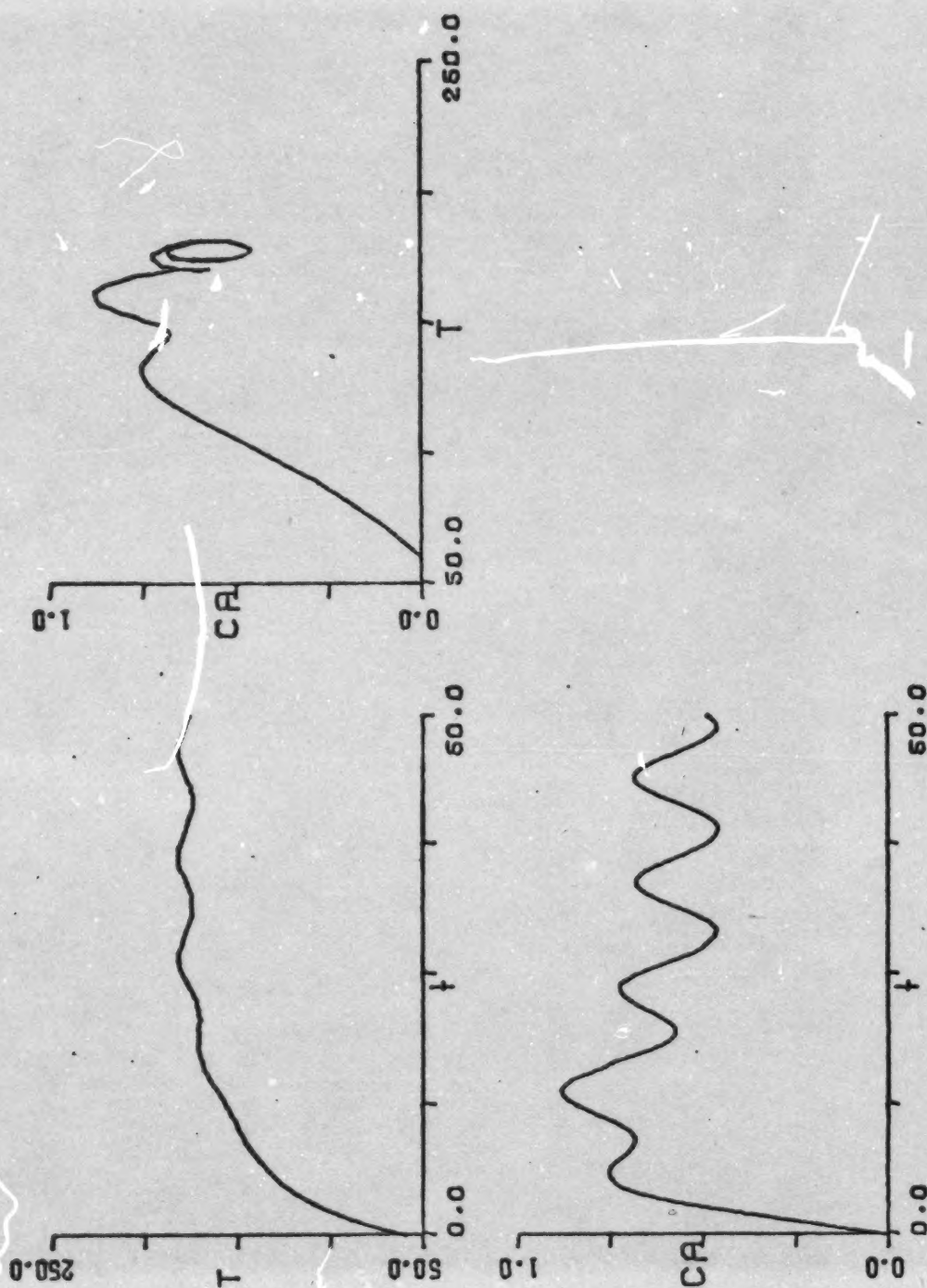


Figure 5. Graphical Output From CSTR Program

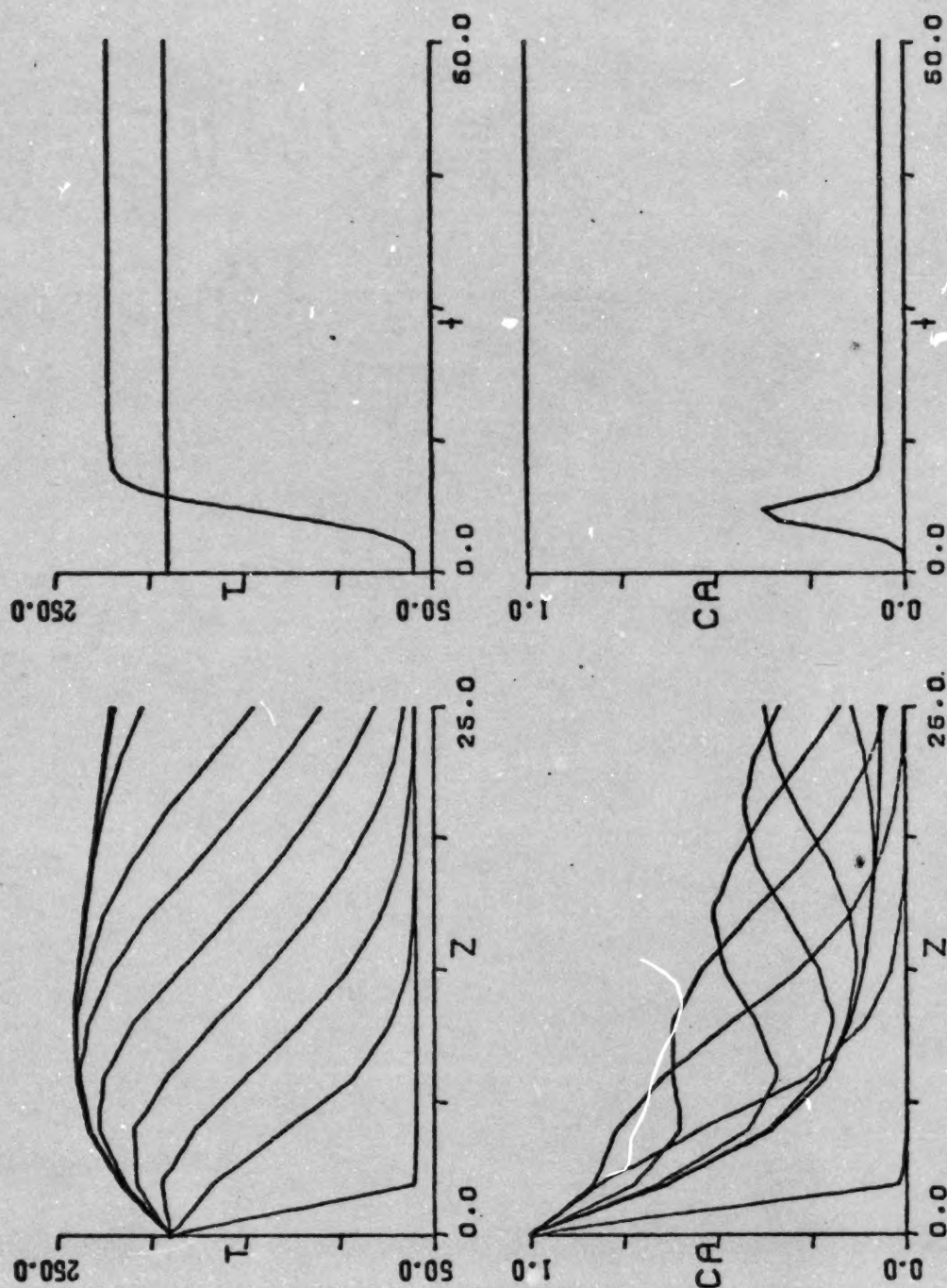


Figure 6. Graphical Output From Tubular Reactor Simulation Program

The total cost of the peripheral hardware is less than \$5,000, and in many cases, only the addition of an X-Y plotter to existing terminals would be required to complete the interactive graphics system. From our results the combination of a printer terminal and X-Y plotter provides an excellent interactive graphics system at a fraction of the cost of more sophisticated graphics terminals. Additional reasons for choosing this combination of peripheral hardware were:

- (1) Highly reliable (low maintenance)
- (2) Easy for students to use and operate without supervision
- (3) Produces complex and unlimited amounts of graphic output per sheet of paper
- (4) Provides the user with hard copies of both the numerical and graphical information
- (5) Least expensive hardware providing interactive graphics.

INSTRUCTIONAL MODULES

The simulation programs were designed to supplement the instruction of process dynamics by providing the students with an opportunity to operate and monitor continuous processes. A series of two hour homework assignments was designed to give one hour of student interaction with the simulation program and approximately an hour of questions based on the output generated during the interactive period. The development of each module is preceded by a list of behavioral objectives and followed by student examinations to evaluate the level of technical learning. In addition, a battery of student attitude tests is given in

an attempt to correlate student characteristics with their success and enjoyment of this type of learning.

Simulation programs and the corresponding instructional modules were presented to the students in the following manner. The CSTR program was introduced to the students through a lecture and group demonstration illustrating how to operate the time-sharing system and the terminal equipment. Following this phase the first module introduced the equations for the model and asked the students to define and compare the nomenclature of the equations to that of the program. The computer was not used in this module. Also, at this time the students are given a step-by-step introduction sheet for operation of the terminal equipment and a condensed instructional manual for the program which includes a definition of user adjustable parameter, instructions for introducing parameter changes into the program, a schematic diagram of the process, and an example of numerical output from the program. After this preparation the students were able to assimilate additional instructional modules as they were needed in supplementing the course.

The instructor may choose the topics and corresponding instructional modules which best supplement the course outline. For a course in process dynamics the instructional modules begin with an exercise in reactor start-up, then a module in which the students investigate the process response to the typical input disturbances, step change, impulse, sinusoidal input, and random input. Next, a module on parametric sensitivity gives the

student a more quantitative feeling for the effect of parameter changes on the system's response. Following these there is a module illustrating the controlled process and one investigating reactor stability and multiple steady state operating conditions. Similar modules have also been developed in conjunction with the tubular reactor program.

The beginning modules are limited in content, user interaction time and complexity of subject matter. The reactor startup module is shown in Appendix I and represents a typical beginning exercise. An example of a more complex and lengthy module, the CONTROLLED CSTR, is shown in Appendix II. These two modules illustrate the type of pedagogical approach taken in using these simulation programs.

In addition to developing and implementing this type of learning aid, extensive efforts are being made to quantitatively assess the teaching and cost effectiveness of the concept. Both before and after the homework assignments, several attitude and other tests developed by the Measurement and Evaluation Center of The University of Texas at Austin were completed by the students. After the completion of the teaching modules, a test was given to evaluate student progress in meeting the behavioral objectives for the module.

Two different attitude measurement tests were administered in order to ascertain the reaction of students to the computer-based instructional modules. A proctor checklist was also used to rate each student as he interacted with the computer and to isolate problem areas with respect

to the instructional module. Information on student interest, anxiety, and problems with the module provided feedback to assist in modifying the modules and writing future modules.

Orientation inventory tests were administered to classify students as to task-, self-, and interaction-orientation in order to determine if these personality characteristics affected the student's performance. Essentially all of the engineers tested were very strongly task oriented. Also, the students were given a questionnaire on orientation toward college. This test indicated that the students were occupationally oriented as opposed to scholarly, social, or individualistic aspects of college life.

To ascertain the students' attitudes toward the computer-based simulations, over sixty multiple choice (five answers) questions were completed by each student in class. Although we do not have space here to describe all the results, five typical questions, and the responses (13 in total) were as follows:

1. Concerning the course material I covered, my feeling toward the material before I started the lessons was
2. My feeling after I had completed the lessons was

	<u>Student Responses</u>	
	<u>1</u>	<u>2</u>
Very favorable	1	1
Favorable	9	12
Indifferent	3	0
Unfavorable	0	0
Very unfavorable	0	0

3. I felt frustrated by the computer-based simulation procedure.
4. In view of the amount I learned, I would say computer-based simulation is superior to traditional instruction.

5. While engaged in computer-based simulation I felt challenged to do my best work.

	<u>Student Response</u>		
	<u>3</u>	<u>4</u>	<u>5</u>
Strongly disagree	0	0	0
Disagree	10	3	6
Uncertain	2	8	1
Agree	1	2	5
Strongly agree	0	0	1

Students were favorably oriented toward the instructional material before the simulations were carried out, and ended up perhaps slightly more favorably oriented after they were completed. Question 4 indicates that the students believed the computer-based instruction to be no better than the lecture-question sessions, but in another essay type question which asked whether or not they would recommend the computer lessons be used again the next time the class was taught, eleven students said yes, and two did not answer the question. Apparently, the students felt the computer sessions were best used as a supplement to the usual class presentation, but should not supplant them. We are now engaged in an analysis of the student responses as related to their grade point averages, SAT scores, class grade, etc.

Student gain of factual subject matter was tested by multiple choice questions and calculational problems. Concrete facts were readily absorbed, but a test of the extension of the students' knowledge to interpret the effect of a new type of simulation (not covered in the computer sessions)

was missed by one-half of the students, so that the objective of providing operating experience for the students was not achieved to the desired degree.

These conclusions and module evaluation results were based on limited test cases. Hopefully, our current testing and evaluation efforts will more thoroughly answer the many questions related to this teaching method.

CONCLUSIONS

Computers in engineering education are commonly used for batch-mode computation, but the application of digital computers in an interactive mode does add a new and powerful tool in teaching abstract and mathematically oriented subjects such as process dynamics and process control. Students' comments strongly indicate a favorable reaction to supplementing courses with this type of computer aided learning. Since the use of computers in education is closely related to costs and available equipment, this work was specifically designed to minimize computational costs and utilize commonly available equipment in so far as possible. We used the least expensive combination of hardware, a teletypewriter and X-Y plotter.

The future of interactive dynamic process simulation on time-sharing systems appears very favorable for processes with short solution times, because the computer response time is short and costs are minimal. However, if the process model expands in computational complexity, the longer

computer response time can nullify the advantage of interactive computation, and for these cases the user would probably revert back to batch-mode computation. The limiting size of process model that can successfully be treated in an interactive mode depends on the computational speed of the time-share computer, the average number of users on line, and the transmission or operating rate in characters per second on the terminal system.

Topics and areas of contribution covered in this work are summarized as follows:

We implemented and tested interactive dynamic simulation, and

1. Investigated two types of algorithms for simulating dynamic processes in chemical engineering (modular and equation solving routines) and concluded that the modular type was preferable for this particular application.
2. Converted Franks' DYFLO routines from a batch-mode algorithm into an interactive simulator.
3. Developed software for and tested various graphics equipment to achieve a practical unit for providing the desired form of interactive graphics at the least cost.

4. Introduced and tested the instructional module concept for using computer-student interaction efficiently.

Extensive evaluation of this work is currently in progress to quantitatively determine the learning and cost effectiveness of this form of instruction.

NOMENCLATURE

$A_{0,1,2}$ Frequency factor for zero, first, or second order kinetics, respectively

A Heat transfer area of the cooling coils, ft^2

C_A Reactant concentration, lb moles/ft^3

C_B Product concentration, lb moles/ft^3

C_p Heat capacity, same for all components, Btu/lb mole

D_{Az} Axial dispersion coefficient, length/time

E Activation energy, Btu/lb mole

ΔH Heat of reaction, Btu/lb mole

L Length of the tubular reactor, ft

R_{A0} Zero order reaction rate, $\text{lb moles/ (ft}^3 \text{ min)}$

R_{A1} First order reaction rate, $\text{lb moles/ (ft}^3 \text{ min)}$

R_{A2} Second order reaction rate, $\text{lb moles/ (ft}^3 \text{ min)}$

R Gas constant, $\text{Btu/ (lb mole}^\circ\text{R)}$

Pe Peclet number, $V_z L/D_{Az}$

q Volumetric flow rate, ft^3/min

Q Heat removal rate through the cooling coil, Btu/min

t time, min

T Temperature, $^\circ\text{F}$ or $^\circ\text{R}$

U Heat transfer coefficient, $\text{Btu/ (ft}^2 \text{ min } ^\circ\text{F)}$

V_z Velocity of fluid in the axial direction, ft/min

V Volume of the continuous stirred tank reactor, ft^3

z Differential distance, ft

SUBSCRIPTS

- A Reactant
- B Product
- c Coolant stream
- i Inlet conditions
- o Outlet conditions
- 1 Inlet feed stream
- 2 Exit streams
- 3 Coolant inlet stream
- 4 Coolant exit stream

GREEK LETTERS

- ρ Fluid molar density, $\text{ft}^3/\text{lb mole}$

REFERENCES

- 1) PACER: Shannon, F. T., and D. R. Frantz, "The PACER System Manual," Dartmouth College, Hanover, New Hampshire, 1966.
- 2) CSMP: "CSMP" - The Continuous System Modeling Program," IBM Canada Ltd., Manual H20-0367.
- 3) FLOWTRAN: Monsanto Company, "An Introduction of FLOWTRAN--Flowsheet Simulation," Computerized Engineering Applications Department, 800 N. Lindbergh Blvd., St. Louis, Missouri, April, 1970.
- 4) CHESS: Motard, R. L., Lee, H. M., Barkley, R. W., and Ingels, D. M., "CHESS-System Guide, (1968), "Tech. Publishing Co., 4375 Harvest Lane, Houston, Texas 77004.
- 5) DYNYSYS: Bobrow, S., A.I. Johnson, and J. W. Ponton, DYNYSYS Manual and Application Studies. McMaster University, August, 1970.
- 6) DYFLO: Franks, R. G. E., Modeling & Simulation in Chemical Engineering. J. Wiley, New York, April, 1972.
- 7) PRODYC: Ingels, D. M. and R. L. Motard, PRODYC, A Simulation Program for Chemical Process Dynamics and Control. University of Houston, RE 4-70. August, 1970.
- 8) REMUS: Ham, P. G., Users Manual. REMUS (Routine for Executive Multi-unit Simulations.) University of Pennsylvania, June, 1970.
- 9) Douglas, J. M., Process Dynamics and Control, Vol 1 & 2, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.
- 10) Himmelblau, D. M., and K. B. Bischoff, Process Analysis and Simulation Deterministic Systems, John Wiley & Sons, Inc., New York, 1968.

APPENDIX I

SIMULATION OF REACTOR START-UP

The object of this phase of the simulation is to investigate the dynamic response of a CSTR during the start-up period. Model CSTR simulates an operating well stirred reactor with heat exchange similar to equipment you would expect to find in a unit operations laboratory. You are able to control or change the flow rate of the inlet stream and the flow rate of the coolant. Assume for the purposes of start-up that these are the only two variables that can be changed.

You start with the reactor initially full of species B at a temperature of 60 °F. All the other initial values of the variables and coefficients are as listed in the CSTR Model Instructions. Note the reactor volume remains constant in the simulation.

After the teletype types ENTER DATA you may introduce initial conditions or accept the default values listed in the user's instructions. Execute program CSTR and begin the simulation by using the default values, i.e., simply type GO. Repeat GO a sufficient number of times to reach the steady state. Compare the teletype print out with the graphs and follow the course of the response. After reaching the steady state, restart the reactor by entering RS and SA each followed by a carriage return.

Repeat the simulation, but increase the reactor flow rate, F1, by a factor of 2.

Again restart the simulation at time zero. This time, however, devise and implement an operating strategy by adjusting the coolant flow rate that will result in a shorter time to reach the same steady state as in the first run. Test your ideas.

QUESTIONS AND CALCULATIONS TO BE COMPLETED DURING OR AFTER THE SIMULATION

- (1) Calculate the residence time for runs 1 & 2.
- (2) What is the effect of increased reactor flow rate on the start-up time?
- (3) Describe your best start-up strategy in less than 200 words.

*Figure 7 on the following page illustrates the graphical results from this module.

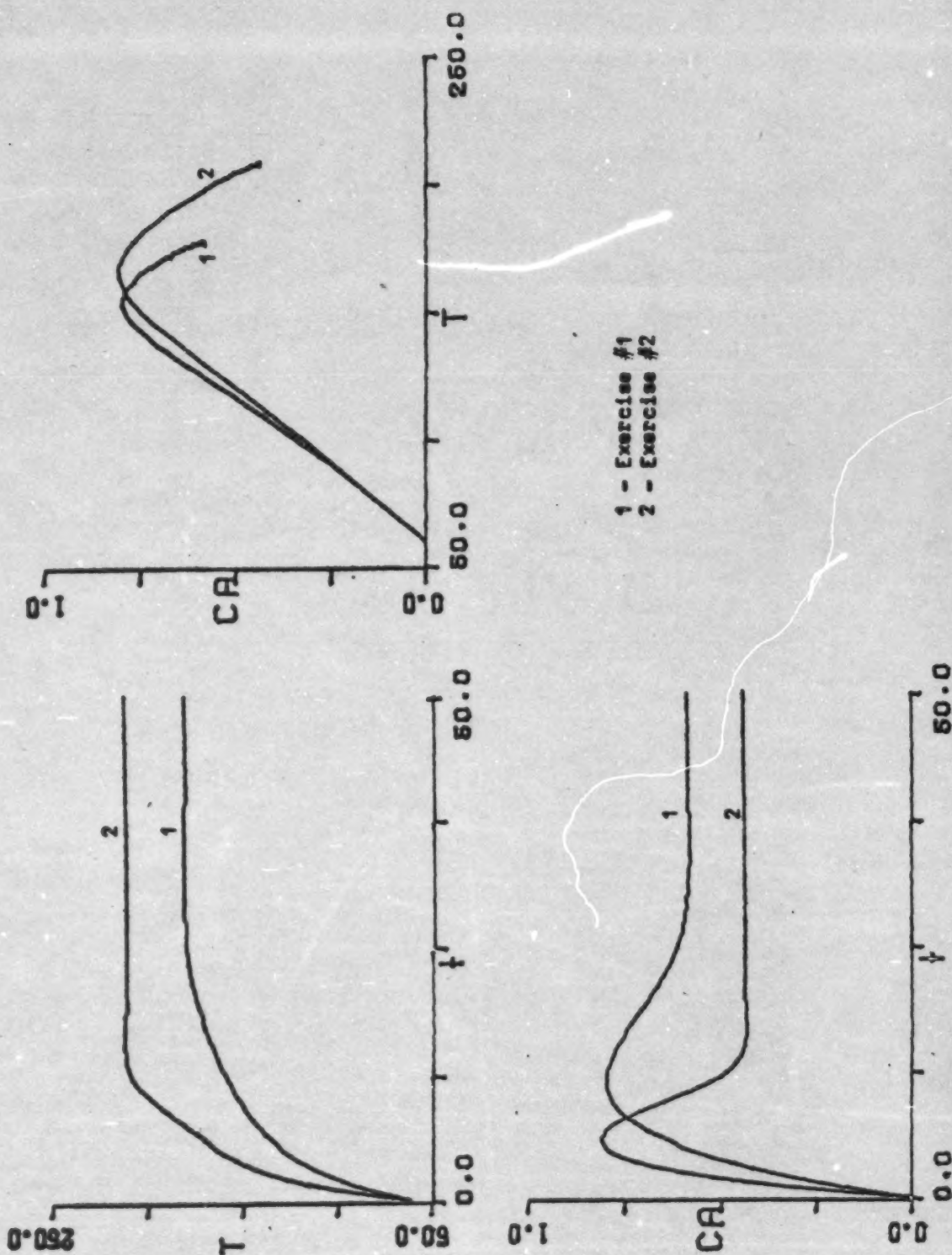


Fig 7. Graphical output from the Reactor Start-up Module.

APPENDIX II

CONTROLLED CSTR

A two mode (proportional-integral) controller has been added to the reactor in order to maintain the reaction temperature at a pre-specified value. The purpose of this module is to demonstrate the dynamic behavior of the controlled continuous stirred tank reactor, CSTR. Of particular interest is how well the specified response can be maintained and what the dynamic response of the reactor is with respect to changes in the controller parameters as well as in the reactor parameters. Included in the CSTR program are the necessary equations describing the operation of the two-mode controller. The user can set the controller into operation by entering the code CN=1.0 followed by a carriage return. Operation of the program remains as described in the CSTR instructions.

DESCRIPTION OF THE CONTROLLER

Suppose the reactor must be operated to meet specific values of C_A and T . We wish to maintain the reactor temperature at a specified value by using a feedback controller to regulate the coolant flow rate. The control loop consists of a temperature sensor, a controller, and a control value. Figure 8 shows the reactor, the input-output streams, and the control loop, the latter by dashed lines representing information flow but not material flow. Figure 9 is a schematic representation of

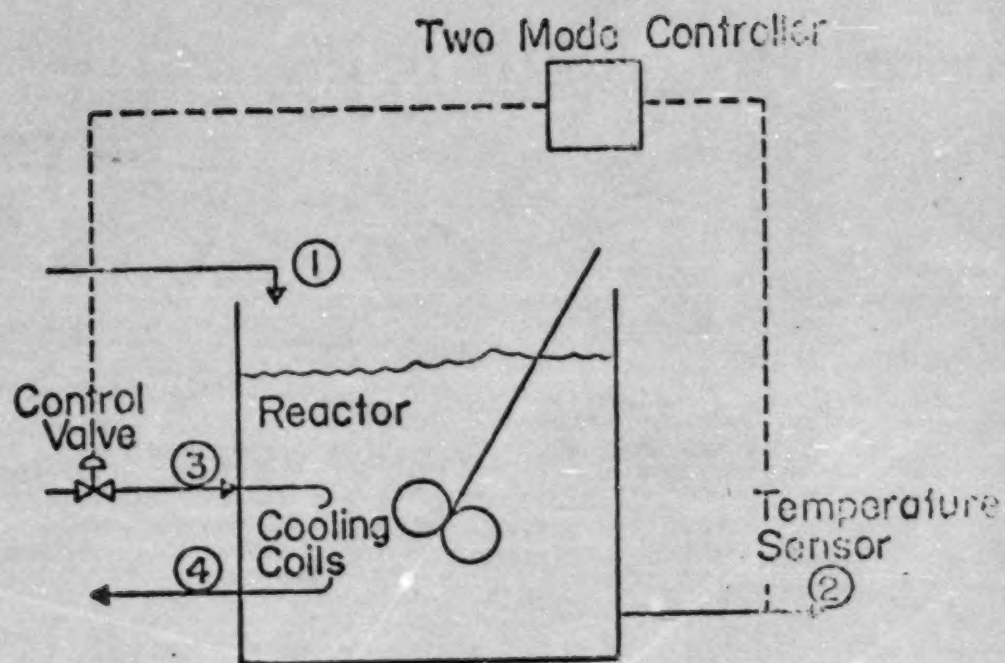


FIGURE 8.

SCHEMATIC DIAGRAM OF THE CSTR SYSTEM

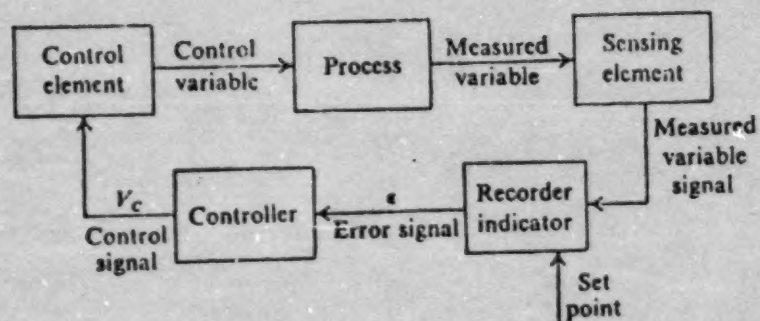


FIGURE 9.* SCHEMATIC DIAGRAM OF THE CONTROL LOOP

*FROM REFERENCE 6.

the control loop. There is a direct relationship between the blocks given in Figure 9 and the actual equipment illustrated in Figure 8 as follows:

<u>Equipment</u> (Fig. 9)	<u>Block Notation</u> (Fig. 8)
1) Temperature sensor (Thermocouple or Thermistor)	Sensing element
2) Controller	Recorder indicator and controller
3) Control valve	Control element
4) Reactor	Process

The three mathematical relations added to the CSTR model that are required to describe the controller operation are listed below. The first equation is an expression for the normalized error, i.e., the deviation between the reactor temperature and the desired temperature or set point.

$$\text{Normalized error} = \text{EPC} = (T_2 - \text{SP}) / \text{SPAN} \quad (1)$$

where

$$T_2 = \text{Reactor temperature, } ^\circ\text{F}$$

$$\text{SP} = \text{Desired temperature, } ^\circ\text{F, called the set point}$$

$$\text{SPAN} = \text{Controller range } ^\circ\text{F}$$

Next the equation describing the controller is

$$\text{CO} = \frac{100}{\text{PB}} (\text{EPC} + \text{CR} \int \text{EPC} \, dt) \quad (\text{AXN}) \quad (2)$$

where

CO = Controller output signal, 0 to 100

PB = Proportional band, 100/gain

CR = Reset, repeats/min., acts as a gain for the integral portion of the controller

EPC = Normalized error

AXN = Controller action, +1 or -1

Finally, the equation relating the controller output to the actual flow rate of coolant is:

$$F3 = 60.0 (CO/100.0) + 0.1 \quad (3)$$

Three parameters may be set or varied by the user, in the equations describing the control loop: namely the proportional band (PB) defined as 100/gain; the reset control (CR) which acts as a gain for the integral portion of the controller; and the set point (SP) i.e., simply the desired reactor temperature. The controller's function is to maintain the desired reactor temperature by properly adjusting flow rate. Qualitatively, when the controller increases the coolant flow rate, the reactor temperature decreases, and conversely, reducing the flow rate produces a high reactor temperature.

CSTR SIMULATIONS

(1) Your first task is to operate the reactor without the aid of the controller but with the objective of maintaining, as an operator would,

an outlet reactor temperature of 160 °F. Log onto the computer and execute the CSTR program. You are to adjust the coolant flow rate, F3, after the ENTER DATA statement in any way you choose in order to maintain the steady-state reactor outlet temperature at 160 °F.

(a) Record the new steady-state coolant flow rate and compare it to the value you will obtain in part (2).

(b) How close did you come to the desired temperature?

Report your final steady-state reactor temperature, T2, and the coolant flow rate, F3.

(2) Continue the simulation from part (1). By introducing the controller for the CSTR, you can determine the correct coolant flow rate corresponding to part (1). Activate the controller and change the set-point to 160 °F. After ENTER DATA, type:

CN = 1.0

SP = 160.0

GO

Compare the observed value of F3 with that reported in part (1). Next, in order to show how the system responds to a change in set-point, introduce a new set-point value of 180 °F and continue the simulation until the steady state is attained. After ENTER DATA, type:

SP = 180.0

GO

Compare the simulation time required for the controller to drive the reactor to a new steady-state temperature with that required by an operator (your assignment in part (1)).

(3) The purpose of the next phase of this assignment is to illustrate the controlled CSTR's response to a change in a reactor input such as the feed temperature. Continue the simulation from part (2), but first increase the inlet temperature to 200 °F. After ENTER DATA, type:

T1 = 200.0

GO

Now you should have a qualitative feeling for the direction of change of the coolant flow rate as set by the controller to maintain a desired temperature. Thus for changes in the inlet temperature, reactant concentration, flow rate, and set-point, you should be able to predict the action taken by the controller. If these relationships are not clear, continue the simulation devising your own experiments.

(4) In parts (2) and (3) you have observed how effective the controller is for maintaining a specific temperature when the reactor is subjected to step changes in various parameters. Now you will see how the controller and system respond to an oscillating or periodic variation in the reactant concentration, C_A . Restart the program and insert new graph paper. After ENTER DATA, type:

RS

Begin this simulation by activating the sinusoidal concentration generator and continue the simulation until a smooth, sinusoidal cycle is observed

for both the outlet concentration and reactor temperature. After ENTER DATA, type:

SC = 1.0

GO

Then activate the controller. After ENTER DATA, type:

CN = 1.0

GO

and observe the reactor temperature and reactant concentration.

(a) What is the effect on the dynamic response of adding the controller?

(5) Repeat part (4), but this time you will introduce a random variation in the reactant inlet concentration. Restart the simulation and again insert new graph paper. After ENTER DATA, type:

RS

Activate the random generator as follows: After ENTER DATA, type:

RX = 1.0

GO

After several periods of simulation, activate the controller. After ENTER DATA, type:

CN = 1.0

GO

(a) Describe the effect of the controller.